




# Leveraging BERT’s Power to Classify TTP from Unstructured Text

Paulo M. M. R. Alves <sup>\*</sup>, Geraldo P. R. Filho <sup>†</sup>, Vinícius P. Gonçalves <sup>‡</sup>  
 Universidade de Brasília, Brasília, Brazil<sup>\*†‡</sup>  
 paulo.magno@aluno.unb.br<sup>\*</sup>, geraldof@unb.br<sup>†</sup>, vpgvinicius@unb.br<sup>‡</sup>

**Abstract**—Tactics, Techniques and Procedures (TTP) are valuable information to cyber-security analysts. However, they are mostly disseminated through unstructured text. This work presents a proposal for tackling this problem by using BERT models, a state-of-the-art approach in Natural Language Processing. We investigate the effect of some chosen hyperparameters on the fine-tuning of the models. MITRE’s example sentences are used to train (fine-tuning step) eleven BERT models. The purpose is to find the best model and the finest combination of hyperparameters for the task of classifying TTPs according to the ATT&CK framework. As a result, we observed that the best models presented an accuracy of 82.64% and 78.75% on two datasets tested, demonstrating the potential of the application of BERT models in the complex task of TTP classification. At last, we gather some insights from the misclassified data that help better understand the dataset and how the models manage and classify the proposed data.

**Keywords**—BERT, TTP, Natural Language Processing, NLP

## I. Introduction

Tactics, Techniques and Procedures (TTPs) are essential information to understand a cyber attack. Though many cyber intelligence feeds focus on Indicators of Compromise (IOCs) and most traditional security solutions are IOC based, this kind of data lacks the attack context TTPs provide. Moreover, many cyber attack campaigns have the ability to mutate IOCs, using customized tools and never previously disclosed infrastructure, effectively escaping IOC based protection [1]–[4]. Thus, understanding the attackers behavior via TTPs becomes essential to defendants.

MITRE has created the ATT&CK framework, a knowledge base of attackers’ behaviors describing known TTPs [5]. Current version of ATT&CK for Enterprise includes 14 Tactics, 191 Techniques, 386 Sub-techniques [6]. However, since TTP information is mostly presented as unstructured text within Cyber Threat Intelligence (CTI) reports, finding and classifying behaviors into those hundreds of possible labels remains a challenge.

To tackle this problem, researchers have resorted to several techniques of Natural Language Processing (NLP) [7]. There are multiple sources of CTI reports and this information overload renders impractical for the defendant to manually peruse every report to extract TTPs [1] [4] [8]–[10]. Automation is paramount [11] [12] and many recent studies have combined machine learning techniques with NLP [13] [14].

Conneau et al [15] work on universal representation of sentences demonstrated that, much like the success it had in computer vision, transfer learning was also suitable to NLP tasks. Vaswani et al [16] proposed the Transformer architecture, a model based solely on self-attention mechanism to represent inputs and outputs.

Leveraging the transfer learning technique and the Transformer model, Devlin et al [17] proposed BERT (Bidirectional Encoder Representations from Transformers). This representation models use two steps, pre-training and fine-tuning, to achieve state-of-the-art results on a variety of NLP tasks, including text classification. Prottasha et al [18] tested different representation schemes (Word2Vec, GloVe, FastText and BERT) and demonstrated that an adequately fine-tuned BERT outperforms other approaches in many NLP tasks, particularly sentiment analysis.

In spite of the evolution of the NLP field with the application of machine learning models, cybersecurity has not fully benefitted from these advances [19]. This research goal is to find the best BERT model and the finest hyperparameters for the task of mapping TTPs to MITRE ATT&CK framework. To the best of our knowledge, our work is the first to employ BERT to this cyber text classification problem. The key contributions of this research are: a) we apply state-of-the-art BERT Transformer architecture to address the challenge of classifying sentences into 253 of the most common attack techniques and sub techniques tabulated in the MITRE matrix; b) we conduct an experimental sweep on different combinations of selected hyperparameters for fine-tuning and evaluate their correlation with performance; c) we identify the best settings and the best

BERT models for unstructured text TTP classification task.

The rest of the paper is structured as follows. Section II provides a background on some related work and the techniques used. In Section III, we discuss the methodology and implementation of our solution. Following that, we present and discuss the results of our approach in Section IV. Finally, in section V, we submit our conclusions and consider future work that could derive from this.

## II. Related Works

In the cybersecurity field much of the NLP research concentrates on extracting IOCs from unstructured text [8] [20], mining social media [21] [22] or amass other cyber security related data [23] [10] [12]. One of the main difficulties in using NLP techniques in the cyber domain is the lack of consistent and annotated datasets [24]. This dearth of data stymies further TTP text classification research [1] [8] [9] [13] [14] [25]. Legoy et al [8] test multiple text representation methods with different multi-label classification models. The best performance was achieved by the TD-IDF bag-of-words text representation method used in conjunction with a Linear SVC classifier.

Husari et al [9] presented TTPDrill, an approach employing TF-IDF method with a version of BM25 information retrieval algorithm. TTP-Drill claims to achieve averages of 84% precision and 82% recall. However, later studies challenge that claim [8] [4]. TTPDrill’s researchers later presented ActionMiner [26], an approach which employs the concepts of entropy and mutual information (from Information Theory) on top of some basic NLP techniques. Other similar works seek for threat actions using a variety of techniques [12] [27]. However, neither ActionMiner nor those other researches map the threat actions to a standardized TTP framework, such as MITRE ATT&CK.

Ayoade et al [4] uses a bias correction method and confidence propagation to predict kill chain phases, tactics and techniques present at a CTI report. KMM, KLIEP and arulSIF methods are applied to estimate the importance weight, which is passed to a SVM classifier. You et al [1] propose the Threat Intelligence Mining (TIM) framework, developing the Threat Context Enhanced Network (TCENet). This tool groups sets of three continuous sentences as candidate text for TTP discovery. The authors limited the scope of their research to the five most popular techniques and one tactic from ATT&CK, obtaining good results.

Another relevant work is TRAM (Threat Report ATT&CK Mapper), done by MITRE Corporation. This tool applies Logistics Regression to predict techniques

for sentences. It uses MITRE Procedure Examples for each technique as a training dataset. However, each proposed classification should be manually reviewed by a human analyst [28].

Following the recent trend of merging machine learning with traditional NLP techniques, we use BERT models to match cyber text sentences to its corresponding TTP in ATT&CK framework. We work on sentence level, similar to [1], [9] and [28], not on document level like [4] and [8].

## III. Methodology

This section describes our strategy to identify TTPs from sentences, classifying them according to MITRE ATT&CK. Figure 1 presents an overview of our proposal. The first step is to prepare the data, by tokenizing and encoding the sentences to BERT format. Next, we split the sentences into training, validation and testing datasets using a stratification procedure to ensure representation of all techniques in each dataset. After that, we fine-tune 11 BERT models using the training and validation datasets using initial hyperparameters derived from literature. We also perform a hyperparameter search to seek for optimization. At last, we use our fine-tuned models, with both initial and optimized hyperparameters, to perform the classification task on the testing dataset and on a manually annotated dataset and analyze the results.

The next subsections describe the data used and further detail each of these steps.

### A. MITRE’s dataset

Since the public release of ATT&CK framework in 2015, MITRE maintains a curated knowledge base with information extracted from CTI reports and manually annotated [3] [5]. This repository, at the time of this writing, contains 10360 sentences (called “procedure examples”) illustrating cyber-attack techniques (and subtechniques).

Not all techniques are illustrated with these procedures examples though. Of the 576 techniques, 466 have at least one example. We chose to work with the most commonly seen techniques, the ones with at least 5 sentences, since the machine learning approach needs examples to learn. That reduces our scope to 253 techniques and subtechniques. With that definition, our dataset is comprised of 9909 of the 10360 sentences, effectively using 95.6% of the examples.

In MITRE’s repository, each sentence is labeled with one technique or subtechnique. Considering this specificity, we tackled the problem of TTP classification using a multiclass approach. Each of the 253 techniques and subtechniques will constitute one class.

### B. Tokenization, Encoding and Splitting

Another peculiarity of this dataset is that it is highly imbalanced. While we limited the smaller classes to 5 elements, the largest one has 371. Class imbalance is a common issue in textual datasets [29]. Some studies demonstrate, however, that BERT deals competently with imbalanced dataset and augmentation strategies normally have limited to no effect on performance [24] [30]–[32]. Nonetheless, as our dataset contain some very small classes, we want to avoid the extreme case where no sentence of one class falls into the training dataset when randomly sampling.

After tokenizing and BERT-ready encoding each sentence, we split the data into training, validation and testing datasets with a 60:20:20 ratio. We use a stratified sampling strategy to ensure that even the smaller classes (5 samples) are represented in each of the three datasets. After the splitting, we have training, validation and testing datasets comprising 5945, 1982 and 1982 samples, respectively.

### C. Models tested

We start by running a simple baseline model comprised of a combination of TF-IDF bag-of-words tokenization scheme with Linear Regression classification model. The datasets are then processed using eleven versions of BERT. Nine of those are of known pre-trained BERT models: BERT base cased, BERT base uncased, BERT Large cased, BERT Large uncased, RoBERTa base, RoBERTa large, DistilRoBERTa, DistilBERT uncased and DistilBERT cased. We also run two models pretrained on a corpus of cybersecurity text: SecBERT and SecRoBERTa. The domain specific pretraining includes data from CTI reports and has its own enhanced vocabulary.

### D. Initial hyperparameters

To define initial hyperparameters for fine-tuning, we resort to some previous works as well as some experimentation. Devlin et al [17] suggest some specifications

that should work fine across different tasks: batch sizes of 16 or 32; learning rates of 5e-5, 3e-5 or 2e-5; training for 2, 3 or 4 epochs. In their article about fine-tuning, Sun et al [33] found batch size of 24, learning rate of 2e-5, maximum sentence length of 128 and 4 epochs training to be reasonable settings for most tasks. Jeawak et al [34] fine-tuned for domain specific classification during 4 epochs with learning rate of 2e-5, batch size of 16 and maximum length of 256.

Analysing MITRE’s sentences, we find that the largest one is 136 tokens long. We set the max\_length parameter to 256 to support that and account comfortably for longer sentences in CTI reports. Every sentence is padded or truncated to this fixed limit. Considering our generous max\_length and the high number of classes we have (253 labels), we set the batch size to 16, to ensure our training steps will fit into GPU memory available. Our large number of classes should also make for slower convergence, so we fine-tune every model for 30 epochs. We use the suggested 2e-5 learning rate in our initial settings.

### E. Hyperparameters search

We conduct a brief hyperparameter search to investigate how some hyperparameters affect accuracy on a model and seek possible improvements in our settings. We chose to examine learning rate and batch size parameters. Learning rate is, arguably, the most important parameter to fine-tune [35]–[37], whereas batch size effect on accuracy is yet to be fully understood [38]–[40].

Considering all the aforementioned theoretical and empirical evidence, we opt to test learning rates of 1e-4, 5e-5, 2e-5 and 1e-5 and batch sizes of 8, 16, 24 and 32. We sweep through all 16 possible combinations. After discovering the best settings, we run the models again with these values to observe the impact on performance.

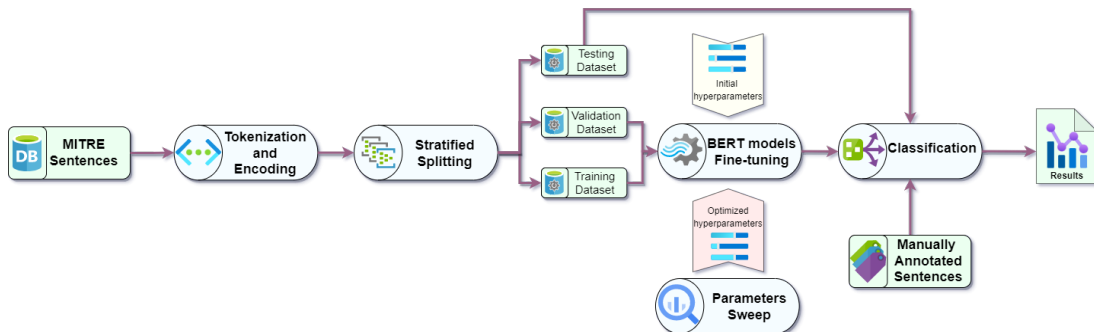


Figure 1: Summarized workflow for TTP classification proposal.

### F. Manually annotated sentences

We also examine how the fine-tuned models performs in data not from MITRE’s repository. We make predictions on an annotated dataset consisting of 80 sentences manually extracted from CTI reports for this inference task. Those sentences were retrieved from 18 reports from 15 different source organizations to ensure language and writing style variety.

## IV. Results and Discussion

We use accuracy to evaluate the performance. Since there is some imbalance but no strong dominance of classes in MITRE’s base (the largest class comprises only 3.58% of all samples) and our classification problem does not value one class over other, accuracy provide good insights into overall performance [41]. This metric is defined as the correct predictions divided by the total predictions, as in Equation 1<sup>1</sup>:

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (1)$$

The baseline TF-IDF/Linear Regression model provides an accuracy of 0.6051 on testing dataset and 0.4771 on inference dataset. We fine-tuned the eleven versions of BERT on MITRE’s knowledge base. Fig. 2 below illustrates the models behavior by showing accuracy and training loss curves for BERT Base Uncased trained for 30 epochs:

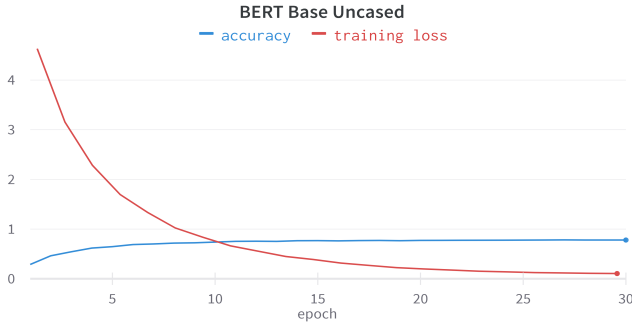


Figure 2: BERT Base Uncased accuracy and training loss curves (learning rate = 2e-5 and batch size = 16).

The curves present the expected machine learning pattern, with accuracy plotting an ascending curve and training loss decaying, demonstrating the model trained correctly. Table I presents the accuracy obtained against the testing dataset, which consists of a detached subset of the MITRE base (not seen by the model during training), and against the set of annotated sentences we use for inference.

<sup>1</sup>TP=True Positive; TN=True Negative; FN=False Negative; FP=False Positive.

Table I: BERT models accuracy on testing and inference datasets with initial hyperparameters.

Models	Testing Dataset	Inference Dataset
BERT Base Uncased	0.7719	0.6375
BERT Base Cased	0.7906	0.7125
BERT Large Uncased	0.8143	0.7250
BERT Large Cased	0.8032	<b>0.7875</b>
RoBERTa Base	0.7951	0.7000
RoBERTa Large	<b>0.8264</b>	0.7750
DistilRoBERTa Base	0.7931	0.6500
DistilBERT Base Uncased	0.7840	0.7125
DistilBERT Base Cased	0.7729	0.6750
SecBERT	0.7830	0.7000
SecRoBERTa	0.7633	0.7000

The best performing models were RoBERTa Large, with an accuracy of 0.8264 on the testing dataset, and BERT Large Cased, with an accuracy of 0.7875 on the inference dataset. The three Large models had the three best accuracy results for both datasets. This result is expected because the size of the BERT pre-trained model affects performance, albeit not drastically [17].

The table also shows that predictions performance on the inference data is lower than on the testing dataset. We assess this difference is due to some longer, more complex sentences in CTI reports than in MITRE’s set of example procedures. Different organizations and analysts have distinct report conventions and writing styles, resulting in a more heterogeneous dataset.

BERT achieves very good overall performance on the TTP classification problem. Comparing our best models to the baseline TD-IDF/Linear Regression model, we observe an uptick of 22.13 percentage points on accuracy for the testing dataset and 31.04% for the inference data. Comparison between our results and preceding works is somewhat difficult because similarities are limited by distinct initial assumptions.

You et al [1] TCENet classifies TTPs with an average accuracy of 94.1%. However, TCENet testing involved only the five most popular techniques and one tactic. Our study applied BERT models to classify 253 classes of TTPs. Husari et al [9] claim to 84% precision and 82% recall come under significantly different premises. Their approach do not employ machine learning and rely on a previously built ontology, which should be manually rebuilt with every update to the ATT&CK Framework.

In search of optimization on the fine-tuning, we conduct a hyperparameter sweep of 16 possible combinations. Fig. 3 show our results:

For our experimental architecture, we found that learning rate had a positive correlation of 0.670 to the

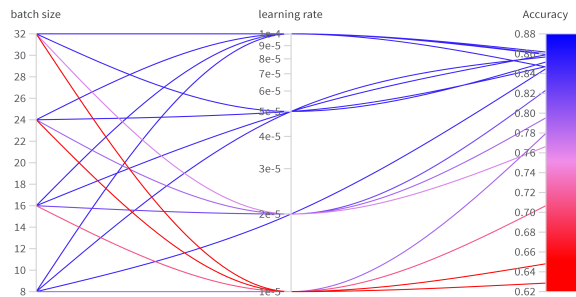


Figure 3: BERT Base Uncased hyperparameters sweep.

target accuracy, meaning higher learning rates should likely produce better results. Batch size, however, did not have a significant correlation (-0.283). The best combination of hyperparameters was batch size of 24 and learning rate of  $1e-4$ . However, when applying this learning rate our experiment faced the catastrophic forgetting effect. This phenomenon consists of the inability of the neural network to retain old information when presented with new one. It is a common problem in machine learning applications for NLP. Higher learning rates setups are more prone to catastrophic forgetting [33] [42]. In our experiment, all three of our Large models, when set with the higher learning rate used ( $1e-4$ ), incurred in catastrophic forgetting.

Table II below show the results obtained by each model when fine-tuned with the optimized hyperparameters.

Table II: BERT models accuracy on testing and inference datasets with optimized hyperparameters (learning rate =  $1e-4$ ; batch size = 24). CF = Catastrophic Forgetting.

Models	Testing Dataset	Inference Dataset
BERT Base Uncased	0.7996	0.7000
BERT Base Cased	0.7840	0.7250
BERT Large Uncased	CF	CF
BERT Large Cased	CF	CF
RoBERTa Base	0.8007	0.6875
RoBERTa Large	CF	CF
DistilRoBERTa Base	<b>0.8012</b>	0.7538
DistilBERT Base Uncased	0.7825	<b>0.7625</b>
DistilBERT Base Cased	0.7936	0.7125
SecBERT	0.7926	0.6750
SecRoBERTa	0.7845	0.7000

Comparing these results to the initially obtained (Table 2) a slight improving tendency is noted in the eight models that did train. The “distilled” models (DistilBERT Cased and Uncased and DistilRoBERTa) showed the highest improvement. Nevertheless, none of these models improved enough to topple the performance achieved by the Large models with our initial empirical parameters. Best accuracies for testing and inference

dataset was obtained with RoBERTa Large (0.8264) and BERT Large Cased (0.7875), respectively, using a learning rate of  $2e-5$  and batch size of 16.

At last, we manually examine the misclassifications from our models. This brief qualitative assessment aims at better understanding where is the model making mistakes and discerning some of the reasons behind those errors. Table III shows a few selected misclassified sentences along with the predicted and correct labels:

Sentence 1 illustrates an interesting yet uncommon case: the predicted label is more precise than the manually annotated one. Both labels are similar: the predicted one is a technique and the annotated is a subtechnique of it. The sentence alone does present enough elements so that the machine learning system could classify it into the subtechnique, though the analyst might have done it considering some context outside of the data in the base.

Sentence 2 reports a case in which both the correct and the predicted label are subtechniques of the same technique. We also observe that the attacker mixed both subtechniques, making both labels correct. Sentences 3 and 4 present a common situation: both labels are adequate, though, in this case, they represent different techniques. This happens because some sentences actually portray more than one technique or subtechnique.

Sentence 5 goes a little further: not only both labels are correct, but there is arguably a third possibility: Exfiltration Over C2 Channel (T1041). This shows that, though MITRE’s database organization leads to a multiclass modeling, the data itself also allow for a multilabel approach.

## V. Conclusions and Future Work

Engineering cyber resilience requires that the best security models are employed to alleviate the burden of cyber analysts [10]. Our work contributes to this intent by harnessing the power of NLP state-of-the-art BERT architecture to the manually cumbersome TTP classification problem. We used MITRE’s labeled sentences base and ran 11 different BERT models, obtaining 82.64% accuracy on test dataset with RoBERTa Large model and 78.75% on inference dataset with BERT Large Cased.

We investigated learning rate and batch size hyperparameters effects on accuracy for potential optimization. Using these optimized settings, we run our pre-trained models again. Low correlation between batch size and accuracy was found. Learning rate, on a different note, produces some improvement on accuracy. The tradeoff is assuming the risk of possibly incurring

Table III: Example of misclassified sentences with correct and predicted labels.

Sentence	Correct Label	Predicted Label
1 MuddyWater has performed credential dumping with LaZagne.	OS Credential Dumping: Cached Domain Credentials (T1003.005)	OS Credential Dumping (T1003)
2 SHIPSHAPE achieves persistence by creating a shortcut in the Startup folder.	Boot or Logon Autostart Execution: Shortcut Modification (T1547.009)	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder (T1547.001)
3 MobileOrder has a command to upload information about all running processes to its C2 server.	Process Discovery (T1057)	Exfiltration Over C2 Channel (T1041)
4 JPIN can use the command-line utility cacls.exe to change file permissions.	Command and Scripting Interpreter: Windows Command Shell (T1059.003)	File and Directory Permissions Modification: Windows File and Directory Permissions Modification (T1222.001)
5 IcedID can inject itself into a suspended msisexec.exe process to send beacons to C2 while appearing as a normal msi application.	System Binary Proxy Execution: Msisexec (T1218.007)	Process Injection (T1055)

in catastrophic forgetting, which we observed in the Large models using the higher learning rate. We also proceeded a qualitative assessment of the misclassifications, learning that some of the supposed errors were actually reasonable predictions but not accounted for due to the multiclass nature of the annotated dataset.

Our proposed work shows that BERT is a powerful tool to help automate mapping of sentences to MITRE ATT&CK TTP framework. To the best of our knowledge, our work is the first to use BERT transformers architecture to the cyber domain specific TTP text classification problem. In the future, it is possible to extend this approach to a multilabel modeling, tackling the issue of longer sentences with multiple TTP descriptions. Also, further investigation can be done into the effects of different parameters on accuracy or other metric of choice.

### References

- [1] Y. You, J. Jiang, P. Jiang, Zhengwei adn Yang, B. Liu, H. Feng, X. Wang, and N. Li, "Tim: threat context-enhanced ttp intelligence mining on unstructured threat data," *Cybersecurity*, vol. 5, no. 3, February 2022.
- [2] Z. Zhu and T. Dumitras, "ChainSmith: Automatically learning the semantics of malicious campaigns by mining threat intelligence reports," in *2018 IEEE European Symposium on Security and Privacy (EuroSP)*, Springer, Ed. London, UK: IEEE, 2018, pp. 458–472.
- [3] B. E. Strom, J. A. Battaglia, M. S. Kemmerer, W. Kupersanin, D. P. Miller, C. Wampler, S. M. Whitley, and R. D. Wolf, "Finding cyber threats with att&ck-based analytics," The MITRE Corporation, Technical Report, 2017. [Online]. Available: <https://www.mitre.org/publications/technical-papers/finding-cyber-threats-with-attck-based-analytics>
- [4] G. Ayode, S. Chandra, L. Khan, K. Hamlen, and B. Thuraisingham, "Automated threat report classification over multi-source data," in *IEEE 4th International Conference on Collaboration and Internet Computing*, IEEE, Ed. Philadelphia, PA, USA: IEEE, 2018.
- [5] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," The MITRE Corporation, Tech. Rep., 2020. [Online]. Available: [https://attack.mitre.org/docs/ATTACK\\_Design\\_and\\_Philosophy\\_March\\_2020.pdf](https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf)
- [6] T. M. Corporation, "Updates - april 2022," April 2022. [Online]. Available: <https://attack.mitre.org/resources/updates/>
- [7] S. M. Zahiri and A. Ahmadvand, "Crab: Class representation attentive bert for hate speech identification in social media," October 2020. [Online]. Available: <https://arxiv.org/abs/2010.13028>
- [8] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated retrieval of att&ck tactics and techniques for cyber threat reports," University of Twente, Enschede, Netherlands, Essay, November 2019.
- [9] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttp-drill: Automatic and accurate extraction of threat actions from unstructured text of cti sources," in *ACSAC 2017: Proceedings of the 33rd Annual Computer Security Applications Conference*, ser. ACSAC '17. New York, NY, USA: Association for Computing Machinery, December 2017, p. 103–115.
- [10] P. Ranade, A. Piplai, A. Joshi, and T. Finin, "Cybert: Contextualized embeddings for the cybersecurity domain," in *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, Ed., December 2021, pp. 3334–3342.
- [11] Y. Harel, I. B. Gal, and Y. Elovici, "Cyber security and the role of intelligent systems in addressing its challenges," in *ACM Transactions on Intelligent Systems and Technology (TIST) - Special Issue: Cyber Security and Regular Papers*, vol. 8, no. 4. New York, NY, USA: Association for Computing Machinery, May 2017.
- [12] Y. Ghazi, Z. Anwar, R. Mumtaz, S. Saleem, and A. Tahir, "A supervised machine learning based approach for automatically extracting high-level threat intelligence from unstructured sources," in *2018 International Conference on Frontiers of Information Technology (FIT)*. Islamabad, Pakistan: IEEE, 2018, pp. 129–134.
- [13] C. Sauerwein and A. Pfohl, "Towards automated classification of attackers' ttps by combining nlp with ml techniques," July 2018. [Online]. Available: <https://arxiv.org/abs/2207.08478>
- [14] R. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A literature review on mining cyberthreat intelligence from unstructured texts," in *2020 International Conference on Data Mining Workshops (ICDMW)*, Sorrento, Italy, 2020.
- [15] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 670–680.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 30. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010.
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language under-

- standing,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, Minnesota, USA, 2019.
- [18] N. J. Prottasha, A. Sami, M. Kowsher, S. A. Murad, A. K. Bairagi, M. Masud, and M. Baz, “Transfer learning for sentiment analysis using bert based supervised fine-tuning,” *Sensors*, vol. 22, no. 11, May 2022.
- [19] P. Institute, “6th cyber resilient organization study,” IBM Security, Tech. Rep., 2021. [Online]. Available: <https://www.ibm.com/resources/guides/cyber-resilient-organization-study/>
- [20] X. Liao, K. Yuan, X. Wang, Z. Li, L. Xing, and R. Beyah, “Acing the ioc game: Toward automatic discovery and analysis of open-source cyber threat intelligence,” in *CCS '16: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Vienna, Austria: Association for Computing Machinery, October 2016.
- [21] S. R. Vadapalli, G. Hsieh, and K. S. Nauer, “Twitterosint: Automated cybersecurity threat intelligence collection and analysis using twitter data,” in *Proceedings of the International Conference on Security and Management (SAM)*, Athens, Greece, 2018.
- [22] N. Dionísio, F. Alves, P. Ferreira, and A. Bessani, “Cyberthreat detection from twitter using deep neural networks,” in *2019 International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, 2019.
- [23] A. Niakanlahiji, J. Wei, and B.-T. Chu, “A natural language processing based trend analysis of advanced persistent threat techniques,” in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, 2018, pp. 2995–3000.
- [24] M. Tikhomirov, N. Loukachevitch, A. Sirotina, and B. Dobrov, “Using bert and augmentation in named entity recognition for cybersecurity domain,” in *Natural Language Processing and Information Systems. NLDB 2020. Lecture Notes in Computer Science()*, vol. 12089. Saarbrücken, Germany: Springer, Cham, 2020, pp. 16–24.
- [25] T. S. Riera, J.-R. B. Higuera, J. B. Higuera, J.-J. M. Herraiz, and J.-A. S. Montalvo, “A new multi-label dataset for web attacks capec classification using machine learning techniques,” *Computers Security*, vol. 120, June 2022.
- [26] G. Husari, X. Niu, B. Chu, and E. Al-Shaer, “Using entropy and mutual information to extract threat actions from cyber threat intelligence,” in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*. Miami, FL, USA: IEEE Press, 2018.
- [27] K. Satvat, R. Gjomemo, and V. N. Venkatakrishnan, “Extractor: Extracting attack behavior from threat reports,” in *2021 IEEE European Symposium on Security and Privacy (EuroSP)*. Vienna, Austria: IEEE, 2021, pp. 598–615.
- [28] J. Yoder, Sarah; Lasky, “Automating mapping to att&ck: The threat report att&ck mapper (tram) tool,” December 2019. [Online]. Available: <https://medium.com/mitre-attack/automating-mapping-to-attack-tram-1bb1b44bda76>
- [29] C. Padurariu and M. E. Breaban, “Dealing with data imbalance in text classification,” in *Procedia Computer Science. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019*, vol. 159, pp. 736–745.
- [30] H. T. Madabushi, E. Kochkina, and M. Castelle, “Cost-sensitive bert for generalisable sentence classification with imbalanced data,” in *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Hong Kong, China: Association for Computational Linguistics, January 2019, pp. 125–134.
- [31] R. Iikura, M. Okada, and N. Mori, “Improving bert with focal loss for paragraph segmentation of novels,” in *Distributed Computing and Artificial Intelligence, 17th International Conference (DAI 0220). Advances in Intelligent Systems and Computing.*, vol. 1237. L’Aquila, Italy: Springer, Cham, 2020, pp. 21–30.
- [32] R. Oak, M. Du, D. Yan, H. Takawale, and I. Amit, “Malware detection on highly imbalanced data through sequence modeling,” in *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*. London, United Kingdom: Association for Computing Machinery, 2019, pp. 37–48.
- [33] C. Sun, X. Qiu, Y. Xu, and X. Huang, “How to fine-tune bert for text classification?” in *Chinese Computational Linguistics: 18th China National Conference, CCL 2019*. Kunming, China: Springer International Publishing, 2019, pp. 194–206.
- [34] S. Jeawak, L. Espinosa-Anke, and S. Schockaert, “Cardiff university at semeval-2020 task 6: Fine-tuning bert for domain-specific definition classification,” in *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, Barcelona, Spain, December 2020.
- [35] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series) Illustrated Edition*, illustrated ed. The MIT Press, 2016.
- [36] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, “The effect of adaptive learning rate on the accuracy of neural networks,” *International Journal of Advanced Computer Science and Applications*, pp. 736–751, 2021.
- [37] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” in *Neural Networks: Tricks of the Trade*, 2nd ed., G. Montavon, Ed. Berlin, Germany: Springer Berlin Heidelberg, 2012, pp. 437–478.
- [38] N. B. Aldin and S. S. A. B. Aldin, “Accuracy comparison of different batch size for a supervised machine learning task with image classification,” in *2022 9th International Conference on Electrical and Electronics Engineering (ICEEE)*, 2022, pp. 316–319.
- [39] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, “On large-batch training for deep learning: Generalization gap and sharp minima,” in *5th International Conference on Learning Representations, ICLR 2017*, Toulon, France, 2017.
- [40] F. He, T. Liu, and D. Tao, “Control batch size and learning rate to generalize well: Theoretical and empirical evidence,” in *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019, pp. 1143–1152.
- [41] N. Japkowicz and M. Shah, *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, 2011.
- [42] P. Kaushik, A. Gain, A. Kortylewski, and A. Yuille, “Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping,” in *5th Workshop on Meta-Learning at NeurIPS 2021*, Virtual, 2021.